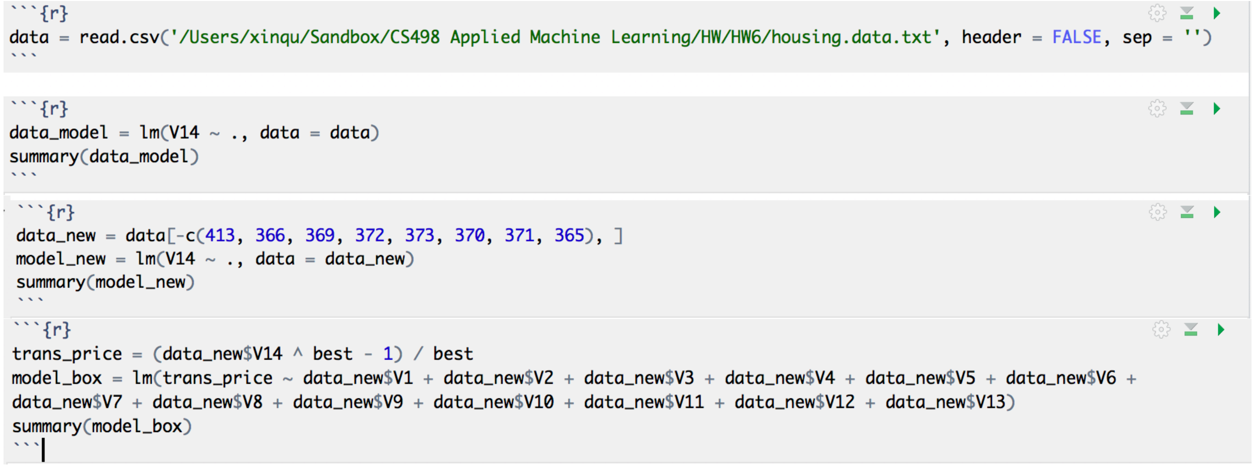
Page 1: code for regression and resulting model

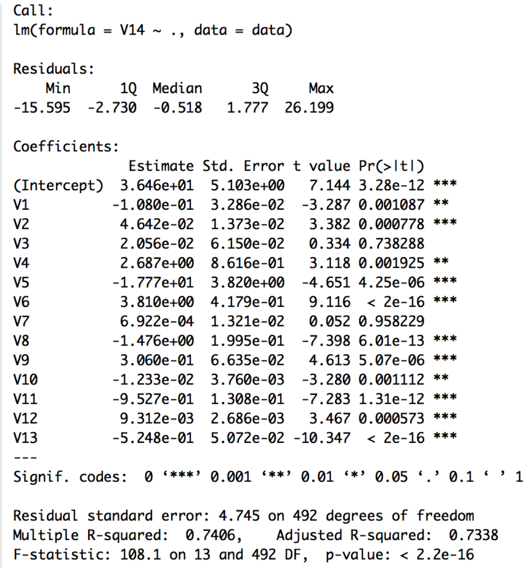


~~~
```{r}
data = read.csv('/Users/xinqu/Sandbox/CS498 Applied Machine Learning/HW/HW6/housing.data.txt', header = FALSE, sep = '')
```
~~~

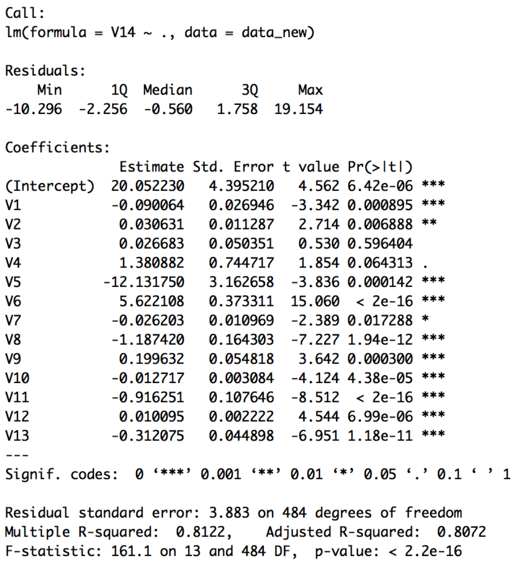
~~~
```{r}
data_model = lm(V14 ~ ., data = data)
summary(data_model)
```
~~~

~~~
```{r}
data_new = data[-c(413, 366, 369, 372, 373, 370, 371, 365), ]
model_new = lm(V14 ~ ., data = data_new)
summary(model_new)
```
~~~

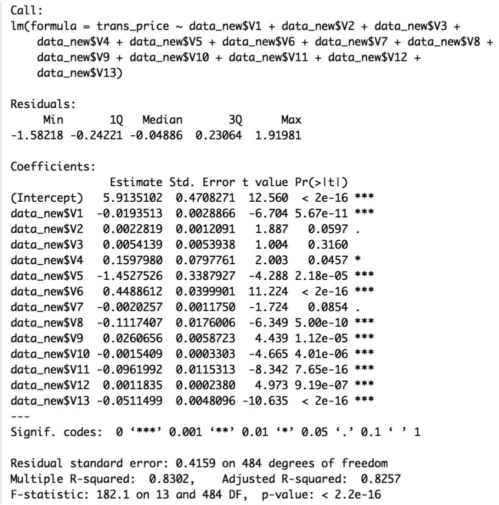
~~~
```{r}
trans_price = (data_new$V14 ^ best - 1) / best
model_box = lm(trans_price ~ data_new$V1 + data_new$V2 + data_new$V3 + data_new$V4 + data_new$V5 + data_new$V6 +
data_new$V7 + data_new$V8 + data_new$V9 + data_new$V10 + data_new$V11 + data_new$V12 + data_new$V13)
summary(model_box)
```
~~~



```
Call:
lm(formula = V14 ~ ., data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-15.595  -2.730  -0.518   1.777  26.199 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  3.646e+01  5.103e+00   7.144 3.28e-12 ***
V1          -1.080e-01  3.286e-02  -3.287 0.001087 ** 
V2           4.642e-02  1.373e-02   3.382 0.000778 ***
V3           2.056e-02  6.150e-02   0.334 0.738288    
V4           2.687e+00  8.616e-01   3.118 0.001925 ** 
V5          -1.777e+01  3.820e+00  -4.651 4.25e-06 ***
V6           3.810e+00  4.179e-01   9.116  < 2e-16 ***
V7           6.922e-04  1.321e-02   0.052 0.958229    
V8          -1.476e+00  1.995e-01  -7.398 6.01e-13 ***
V9           3.060e-01  6.635e-02   4.613 5.07e-06 ***
V10         -1.233e-02  3.760e-03  -3.280 0.001112 ** 
V11         -9.527e-01  1.308e-01  -7.283 1.31e-12 ***
V12          9.312e-03  2.686e-03   3.467 0.000573 ***
V13         -5.248e-01  5.072e-02 -10.347  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.745 on 492 degrees of freedom
Multiple R-squared:  0.7406,	Adjusted R-squared:  0.7338 
F-statistic: 108.1 on 13 and 492 DF,  p-value: < 2.2e-16
```

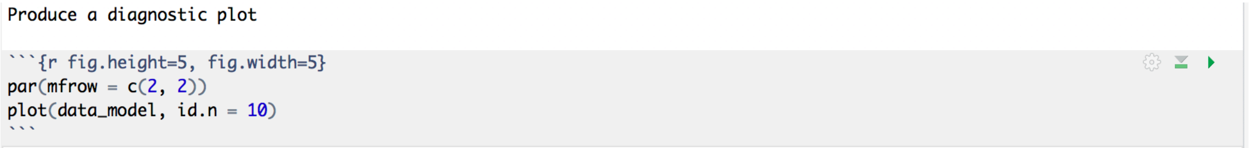


```
Call:
lm(formula = V14 ~ ., data = data_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-10.296  -2.256  -0.560   1.758  19.154 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  20.052230   4.395210   4.562 6.42e-06 ***
V1           -0.090064   0.026946  -3.342 0.000895 ***
V2            0.030631   0.011287   2.714 0.006888 ** 
V3            0.026683   0.050351   0.530 0.596404    
V4            1.380882   0.744717   1.854 0.064313 .  
V5          -12.131750   3.162658  -3.836 0.000142 ***
V6            5.622108   0.373311  15.060  < 2e-16 ***
V7           -0.026203   0.010969  -2.389 0.017288 *  
V8           -1.187420   0.164303  -7.227 1.94e-12 ***
V9            0.199632   0.054818   3.642 0.000300 ***
V10          -0.012717   0.003084  -4.124 4.38e-05 ***
V11          -0.916251   0.107646  -8.512  < 2e-16 ***
V12           0.010095   0.002222   4.544 6.99e-06 ***
V13          -0.312075   0.044898  -6.951 1.18e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.883 on 484 degrees of freedom
Multiple R-squared:  0.8122,	Adjusted R-squared:  0.8072 
F-statistic: 161.1 on 13 and 484 DF,  p-value: < 2.2e-16
```



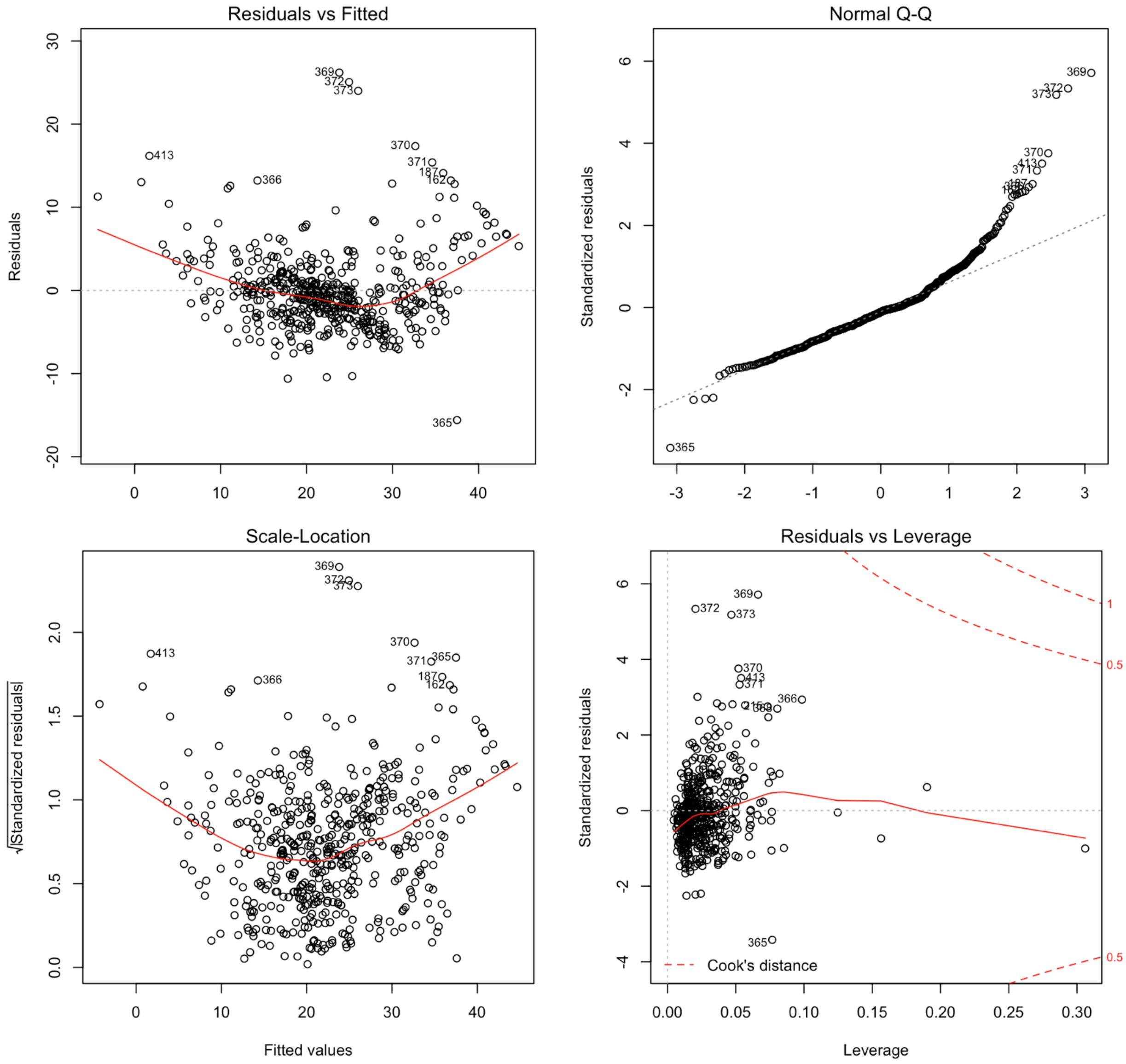
```
Call:
lm(formula = trans_price ~ data_new$V1 + data_new$V2 + data_new$V3 + 
    data_new$V4 + data_new$V5 + data_new$V6 + data_new$V7 + data_new$V8 + 
    data_new$V9 + data_new$V10 + data_new$V11 + data_new$V12 + 
    data_new$V13)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.58218 -0.24221 -0.04886  0.23064  1.91981 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   5.9135102  0.4708271  12.560  < 2e-16 ***
data_new$V1  -0.0193513  0.0028866  -6.704 5.67e-11 ***
data_new$V2   0.0022819  0.0012091   1.887   0.0597 .  
data_new$V3   0.0054139  0.0053938   1.004   0.3160    
data_new$V4   0.1597980  0.0797761   2.003   0.0457 *  
data_new$V5  -1.4527526  0.3387927  -4.288 2.18e-05 ***
data_new$V6   0.4488612  0.0399901  11.224  < 2e-16 ***
data_new$V7  -0.0020257  0.0011750  -1.724   0.0854 .  
data_new$V8  -0.1117407  0.0176006  -6.349 5.00e-10 ***
data_new$V9   0.0260656  0.0058723   4.439 1.12e-05 ***
data_new$V10 -0.0015409  0.0003303  -4.665 4.01e-06 ***
data_new$V11 -0.0961992  0.0115313  -8.342 7.65e-16 ***
data_new$V12  0.0011835  0.0002380   4.973 9.19e-07 ***
data_new$V13 -0.0511499  0.0048096 -10.635  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.4159 on 484 degrees of freedom
Multiple R-squared:  0.8302,	Adjusted R-squared:  0.8257 
F-statistic: 182.1 on 13 and 484 DF,  p-value: < 2.2e-16
```

Page 2: screenshot of diagnostic plot and explanation



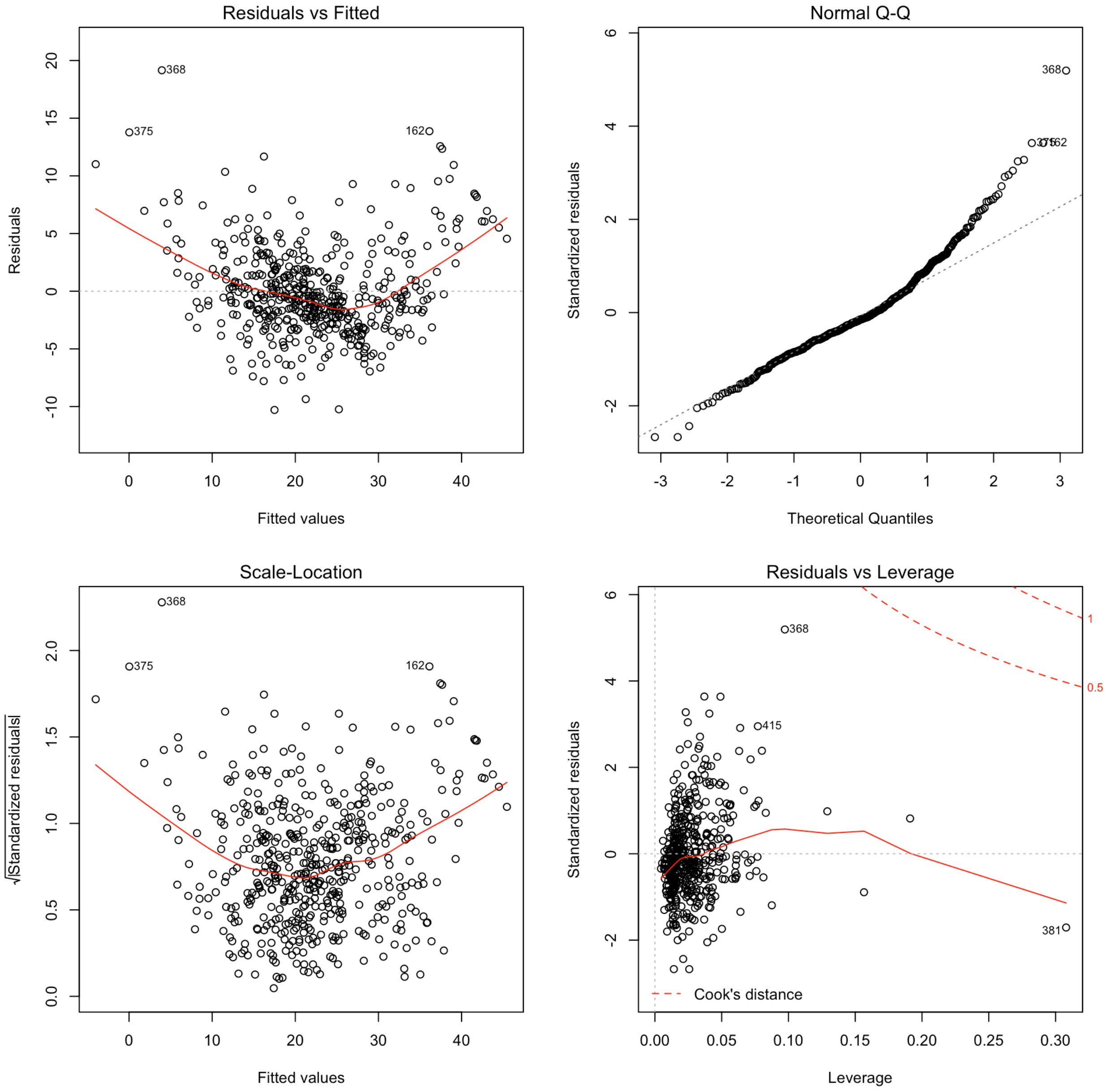
Produce a diagnostic plot

~~~
```{r fig.height=5, fig.width=5}
par(mfrow = c(2, 2))
plot(data_model, id.n = 10)
```
~~~

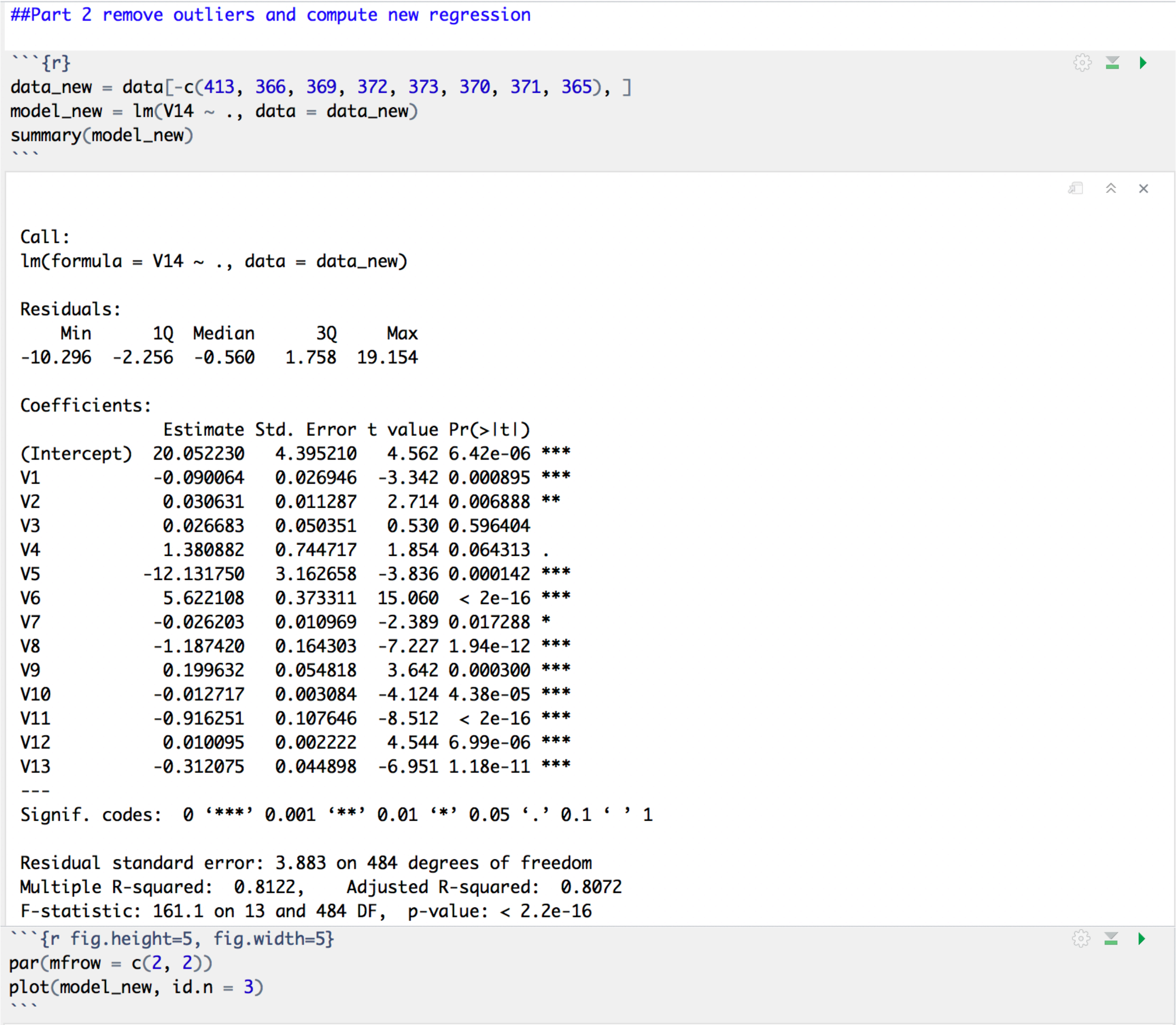


From Residuals vs Fitted plot, it tells that points [413, 366, 369, 372, 373, 370, 371, 187, 162, 365] have great influence on Residuals, since other points spread out equally around the horizontal line which means there is a linear relation between features and the outcome variable if remove these points. Normal Q-Q plot shows if the residuals are normally distributed. From the Normal Q-Q plot, it shows that points [369, 372, 373, 370, 413, 371, 187, 365, 366] are not lined well on the straight dashed line. Scale-Location plot checks the equal variance assumption, it shows that points [413, 372, 371, 366, 373, 365, 369, 370, 187, 162] spread out the line. From Residuals vs Leverage plot, it shows points [372, 373, 369, 370, 413, 371, 366, 215, 158, 365] have the most extreme influence to determine a regression line (with hight value of standardized residuals). The intersection of the listing four point list are [413, 366, 369, 372, 373, 370, 371, 365] (total 8 points) and they are the outliers.

Page 3: screenshot of new diagnostic plot



Page 4: screenshot of code for subproblem 2.

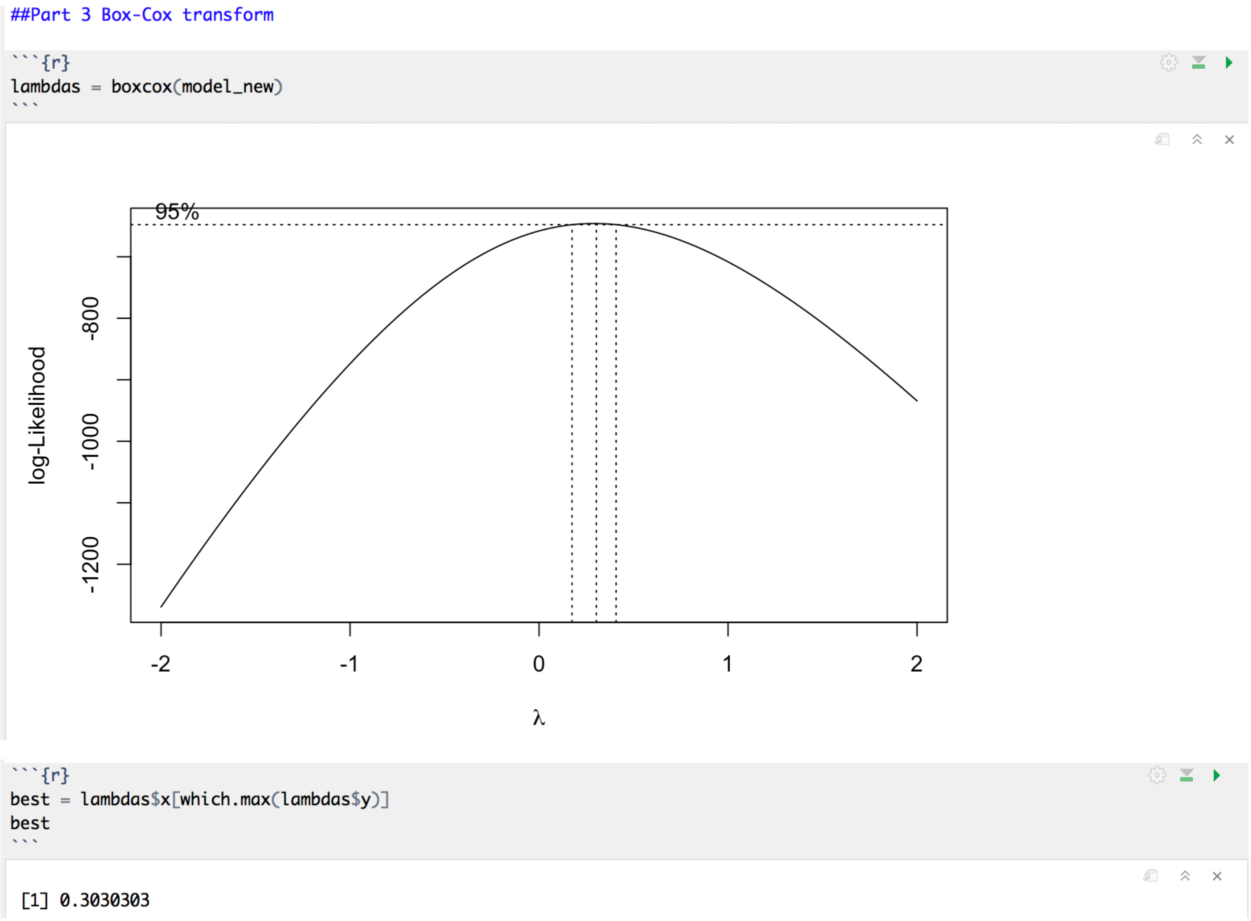


~~~
##Part 2 remove outliers and compute new regression

```{r}
data_new = data[-c(413, 366, 369, 372, 373, 370, 371, 365), ]
model_new = lm(V14 ~ ., data = data_new)
summary(model_new)
```
~~~

```

Call:
lm(formula = V14 ~ ., data = data_new)

Residuals:
    Min      1Q  Median      3Q     Max 
-10.296  -2.256  -0.560   1.758  19.154 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  20.052230   4.395210   4.562 6.42e-06 ***
V1           -0.090064   0.026946  -3.342 0.000895 ***
V2            0.030631   0.011287   2.714 0.006888 ** 
V3            0.026683   0.050351   0.530 0.596404    
V4            1.380882   0.744717   1.854 0.064313 .  
V5          -12.131750   3.162658  -3.836 0.000142 ***
V6            5.622108   0.373311  15.060  < 2e-16 ***
V7           -0.026203   0.010969  -2.389 0.017288 *  
V8           -1.187420   0.164303  -7.227 1.94e-12 ***
V9            0.199632   0.054818   3.642 0.000300 ***
V10          -0.012717   0.003084  -4.124 4.38e-05 ***
V11          -0.916251   0.107646  -8.512  < 2e-16 ***
V12           0.010095   0.002222   4.544 6.99e-06 ***
V13          -0.312075   0.044898  -6.951 1.18e-11 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.883 on 484 degrees of freedom
Multiple R-squared:  0.8122,	Adjusted R-squared:  0.8072 
F-statistic: 161.1 on 13 and 484 DF,  p-value: < 2.2e-16
```

~~~
```{r fig.height=5, fig.width=5}
par(mfrow = c(2, 2))
plot(model_new, id.n = 3)
```
~~~

Page 5: screenshot of Box-Cox transformation and the best value



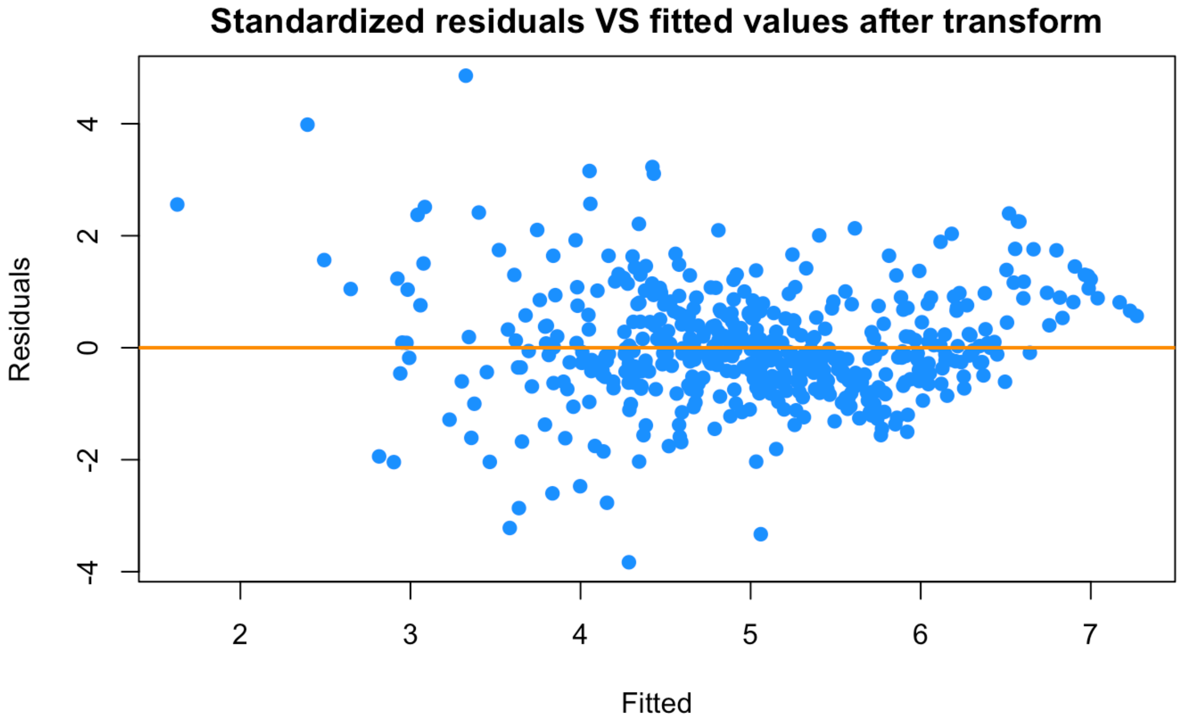
```
##Part 3 Box-Cox transform
```

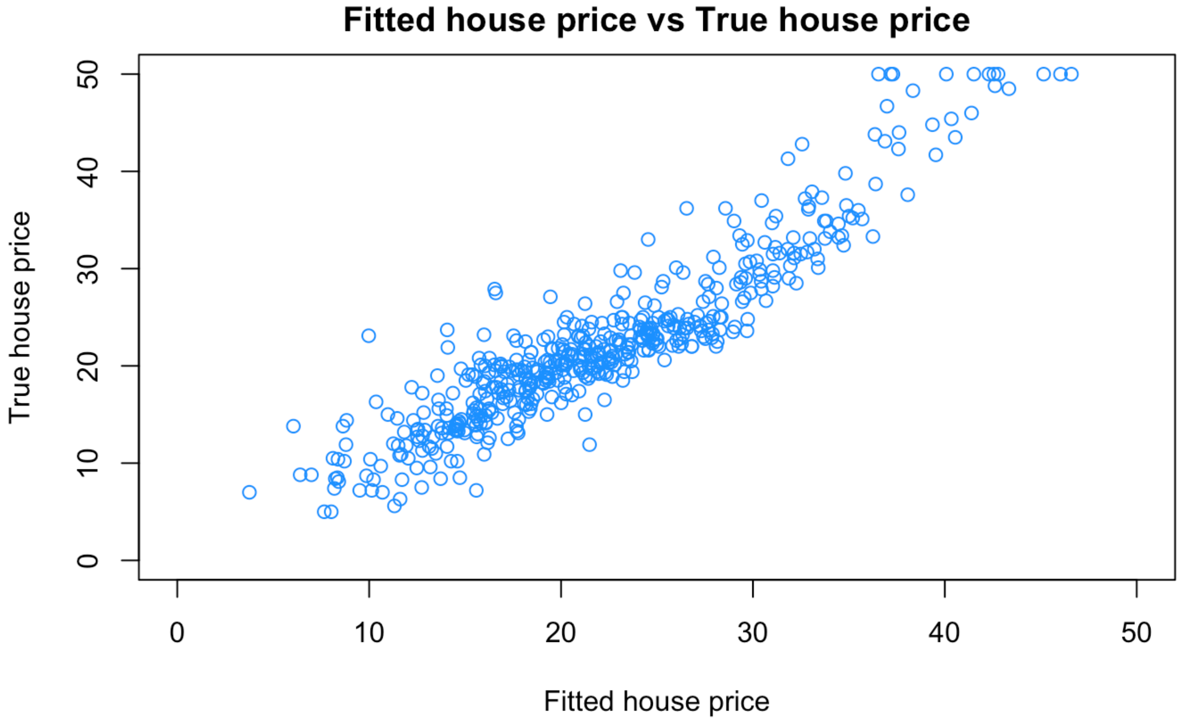
```{r}
lambdas = boxcox(model_new)
```

```{r}
best = lambdas$x[which.max(lambdas$y)]
best
```

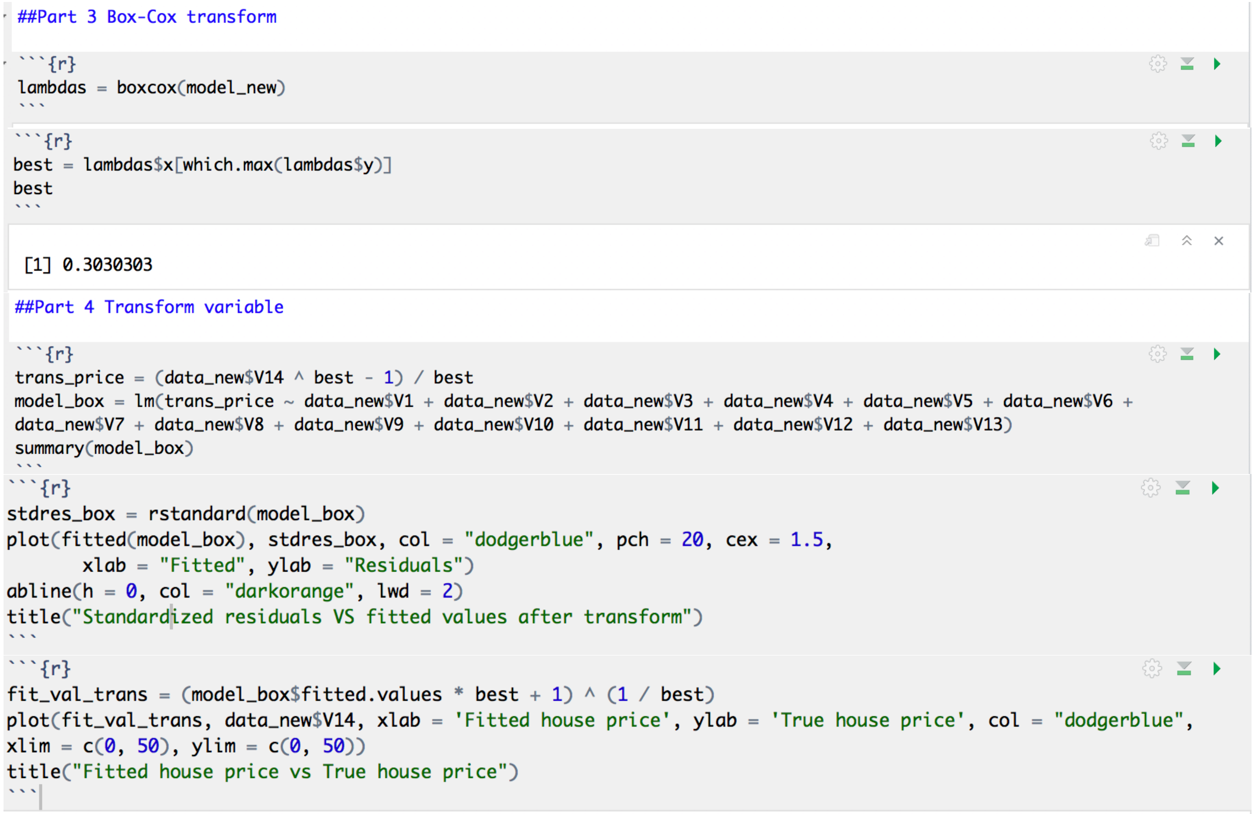
```
[1] 0.3030303
```

Page 6: Standardized residuals of regression after Box-Cox transformation and the plot of fitted house price against true house price





Page 7 code for subproblems 3 and 4



~~~
##Part 3 Box-Cox transform

```{r}
lambdas = boxcox(model_new)
```

```{r}
best = lambdas$x[which.max(lambdas$y)]
best
```

 [1] 0.3030303

##Part 4 Transform variable

```{r}
trans_price = (data_new$V14 ^ best - 1) / best
model_box = lm(trans_price ~ data_new$V1 + data_new$V2 + data_new$V3 + data_new$V4 + data_new$V5 + data_new$V6 +
data_new$V7 + data_new$V8 + data_new$V9 + data_new$V10 + data_new$V11 + data_new$V12 + data_new$V13)
summary(model_box)
```
```{r}
stdres_box = rstandard(model_box)
plot(fitted(model_box), stdres_box, col = "dodgerblue", pch = 20, cex = 1.5,
       xlab = "Fitted", ylab = "Residuals")
abline(h = 0, col = "darkorange", lwd = 2)
title("Standardized residuals VS fitted values after transform")
```
```{r}
fit_val_trans = (model_box$fitted.values * best + 1) ^ (1 / best)
plot(fit_val_trans, data_new$V14, xlab = 'Fitted house price', ylab = 'True house price', col = "dodgerblue",
xlim = c(0, 50), ylim = c(0, 50))
title("Fitted house price vs True house price")
```
~~~